**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

Звіт до Теми №1

Функції та змінні

Під час виконання практичного завдання до Теми №1 було надано варіанти рішення до наступних задач:

**Перетворення рядка**

Необхідно рядок, що має вигляд "abcdefg123" перетворити наступним чином "321gfedcba", вважаючи сталою довжину рядку в 10 символів.

Хід виконання завдання:

Результат виконання програми:



Текст програми:

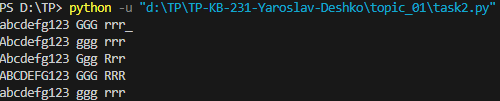
|  |
| --- |
| myStr = "abcdefg123"  print(myStr[::-1]) |

Тестування функцій

Виконати тестування функцій, що працюють з рядками: strip(), capitalize(), title(), upper(), lower().

Хід виконання завдання:

Результат виконання програми:



Текст програми:

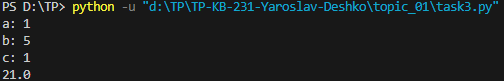
|  |
| --- |
| myStr = "abcdefg123 GGG rrr  "  print(myStr.strip() + "\_")  print(myStr.capitalize())  print(myStr.title())  print(myStr.upper())  print(myStr.lower()) |

Функція пошуку дискримінанта.

Написати функцію пошуку дискримінанту квадратного рівняння.

Хід виконання завдання:

Результат виконання програми:



Текст програми:

|  |
| --- |
| def discriminant(a: float, b: float, c: float) -> float:      return pow(b, 2) - 4\*a\*c  a = float(input("a: "))  b = float(input("b: "))  c = float(input("c: "))  print(discriminant(a, b, c)) |

Звіт до Теми №2

Умовний перехід

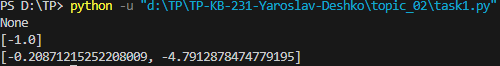
Під час виконання практичного завдання до Теми №2 було надано варіанти рішення до наступних задач:

Пошук коренів квадратного рівняння.

Написати функцію пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту з попередньої теми та умовні переходи.

Хід виконання завдання:

Результат виконання програми:



Текст програми:

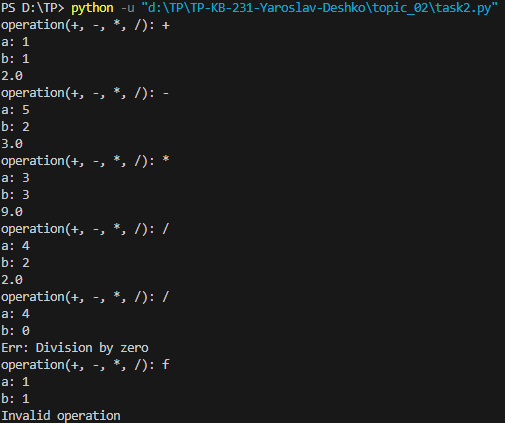
|  |
| --- |
| from math import sqrt  def discriminant(a: float, b: float, c: float) -> float:      return pow(b, 2) - 4\*a\*c  def findRoots(a: float, b: float, c: float):    d = discriminant(a, b, c)    if d == 0:       return [-(b/2\*a)]    elif d > 0:       return [((-b)+sqrt(d))/ 2\*a,               ((-b)-sqrt(d))/ 2\*a ]    return None  def test():    print(findRoots(1, 2, 2)) # D < 0    print(findRoots(1, 2, 1)) # D == 0    print(findRoots(1, 5, 1)) # D > 0      test() |

Калькулятор на **if else.**

Написати програму калькулятор використовуючи **if else** конструкцію. Кожна операція має бути виконана в окремій функції.

Хід виконання завдання:

Результат виконання програми:



Текст програми:

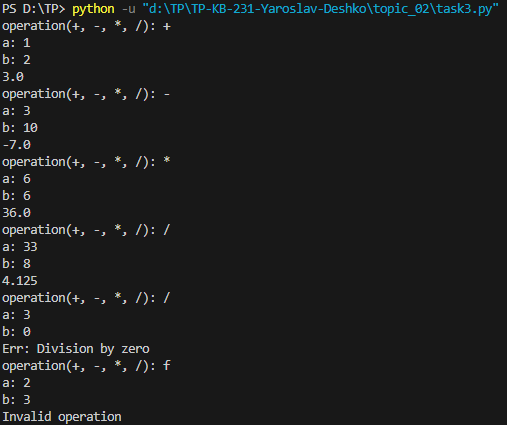
|  |
| --- |
| def calculator():      opr = input("operation(+, -, \*, /): ")      a, b = enterNum()      if opr == "+":         return add(a, b)      elif opr == "-":         return sub(a, b)      elif opr == "\*":         return mul(a, b)      elif opr == "/":         return div(a, b)      else: return "Invalid operation"  def enterNum():      a = input("a: ")      b = input("b: ")      return float(a), float(b)  def add(a, b):      return a+b  def sub(a, b):      return a-b  def mul(a, b):      return a\*b  def div(a, b):      if b == 0 : return "Err: Division by zero"      return a/b  while True:      print(calculator()) |

Калькулятор на **match.**

Написати програму калькулятор використовуючи **match** конструкцію. Кожна операція має бути виконана в окремій функції.

Хід виконання завдання:

Результат виконання програми:



Текст програми:

|  |
| --- |
| def calculator():      opr = input("operation(+, -, \*, /): ")      a, b = enterNum()      match opr:          case "+":              return add(a, b)          case "-":              return sub(a, b)          case "\*":              return mul(a, b)          case "/":              return div(a, b)          case \_:              return "Invalid operation"  def enterNum():      a = input("a: ")      b = input("b: ")      return float(a), float(b)  def add(a, b):      return a+b  def sub(a, b):      return a-b  def mul(a, b):      return a\*b  def div(a, b):      if b == 0 : return "Err: Division by zero"      return a/b  while True:      print(calculator()) |

Посилання на github:

https://github.com/Kaspo02/TP-KB-231-Yaroslav-Deshko

Знімок екрану з посилання на github:

